What else we would love to know

Q. What are the assumptions that you made during the implementation
A. I assumed that both selections.csv and clicks.csv are locally available as I am mounting them as volume

Q. What are the performance characteristics of your implementation?
A. This app is minimalistic implementation, It runs as N threads under gunicorn+flask. It performs well under the load as depicted in load_images folder.

Q. If you could load test it, what do you expect to see in the result?
A. The results are depicted in load_images folder

Q. If you had more time, how would you improve your solution?
A. Many things
   a. Use of async f/w like aiohttp, sanic to serve more requests
   b. Secure the APIs with tokens
   c. Build a caching mechanism for the most frequent user_ids
   d. Kubernetize the app as this is very stateless so that we can scale easily.

Bonus

- What other user insights could we possibly generate from this data?
  - Hotels clicked by users in given region
  - A trend of users' clicked hotels thus we can show him more relevant hotels
  - What hotels are most popular in given region?
  - What hotels are most famous for given period?
  - What amenities are most checked by user at some period?
- If you had to update the data source in real time, how would your solution change?
  - In that case, we need to restart the container as csvs are loaded first thing at startup not for each and every request.
- What comments would you expect when this goes to a code review?
  - No use of blueprints in flask-app
  - Any other uWSGI compliant framework